w.c.s. - Development #35386

throttling sur l'URL du code de suivi

13 août 2019 14:00 - Frédéric Péters

Statut:	Fermé	**Début:**	13 août 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Frédéric Péters	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Pour réduire les possibilités de tests en masse.

Révisions associées

Révision 32f304fd - 13 août 2019 17:13 - Frédéric Péters

misc: add rate limiting to tracking code URL (#35386)

Historique

#1 - 13 août 2019 14:08 - Frédéric Péters

- Fichier 0001-misc-add-rate-limiting-to-tracking-code-URL-35386.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

De manière totalement arbitraire trois requêtes par seconde.

Ça utilise django-rate-limit il me semble que j'avais noté dans un commentaire ailleurs qu'après un rapide tour ça me semblait pas mal, maintenu et packagé (mais dans une ancienne version). J'ai poussé tout à l'heure la dernière version dans nos dépôts.

#2 - 13 août 2019 14:46 - Thomas Noël

Frédéric Péters a écrit :

> De manière totalement arbitraire trois requêtes par seconde.

Mes quelques centimes, en dehors du code qui me semble bon.

3/s me semble un peu bas, aujourd'hui une IP peut être partagée par 10000 personnes en mobile (mais bon, avoir 10 utilisateurs qui cherchent un code de suivi sur leur téléphone et qui seraient sur la même IP dans la même seconde... ok, on n'est pas encore à ce niveau de succès !)

Blague à part je pense plutôt aux agents d'accueil d'une collectivité, tous derrière la même IP. Or il semble bien que django-ratelimit ne gère pas de whitelist, c'est plutôt dommage :/

Tout ça pour dire que 10/s ?

Mais aussi, cela va imposer d'avoir un bon réglage des reverse-proxy... Sur notre SaaS on peut, mais ailleurs ça ne sera pas toujours évident : il faudrait pouvoir débrayer cette limitation dans les mauvaises infra (et dire alors "débrouillez-vous pour faire du ratelimit sur telles URLs, Publik ne peut pas").

Au final je verrais bien une option dans site-options.cfg

```
trackingcode-ratelimit = 10/s
```

histoire de pouvoir la poser à 1000/s sur les instance où on n'arrive pas à avoir l'ip.

#3 - 13 août 2019 15:42 - Benjamin Dauvergne

Thomas Noël a écrit :

> Frédéric Péters a écrit :
>
> > De manière totalement arbitraire trois requêtes par seconde.

> Mes quelques centimes, en dehors du code qui me semble bon.
>
> 3/s me semble un peu bas, aujourd'hui une IP peut être partagée par 10000 personnes en mobile (mais bon, avoir 10 utilisateurs qui cherchent un code de suivi sur leur téléphone et qui seraient sur la même IP dans la même seconde... ok, on n'est pas encore à ce niveau de succès !)
>
> Blague à part je pense plutôt aux agents d'accueil d'une collectivité, tous derrière la même IP. Or il semble bien que django-ratelimit ne gère pas de whitelist, c'est plutôt dommage :/
>
> Tout ça pour dire que 10/s ?
>
> Mais aussi, cela va imposer d'avoir un bon réglage des reverse-proxy... Sur notre SaaS on peut, mais ailleurs ça ne sera pas toujours évident : il faudrait pouvoir débrayer cette limitation dans les mauvaises infra (et dire alors "débrouillez-vous pour faire du ratelimit sur telles URLs, Publik ne peut pas").
>
> Au final je verrais bien une option dans site-options.cfg
>
> [...]
>
> histoire de pouvoir la poser à 1000/s sur les instance où on n'arrive pas à avoir l'ip.

Est-ce qu'on ne devrait pas incrémenter uniquement sur un échec déjà ? Donc :

```
if is_rate_limited(..., increment=False):
   fail()
# search
if not found:
   is_rate_limited(..., increment=True)
```

Ensuite je serai plutôt pour augmenter la période, genre 1000 codes/semaines par IP, ça me parait suffisant et c'est déjà beaucoup plus limitant que 3 par secondes. Le souci c'est que nos clés de cache vont rester une semaine et donc on peut blinder notre memcache d'IP pourries. Pour les agents je dirai qu'il suffit de ne pas utiliser l'IP des gens connectés, mais comme on a pas l'utilisateur au niveau Django avec Quixote j'ai du mal à voir comment faire rentrer ça dans leur API.

Mais donc avec leur API ça donnerait un truc comme ça :

```
        ratelimited = ratelimit.utils.is_ratelimited(
                request=get_request().django_request,
                group='trackingcode',
                key='user_or_ip',
                rate='1000/7d',
                increment=True)
```

ou alors les gens avec au moins un rôle (idem pseudo-code faux puisque je suppose que request.user est un user w.c.s.) :

```
        ratelimited = ratelimit.utils.is_ratelimited(
                request=get_request().django_request,
                group='trackingcode',
                key=lambda group, request: str(request.user.id) if request.user and request.user.roles else re
quest.META['REMOTE_ADDR'],
                rate='1000/7d',
                increment=True)
        if ratelimited:
            raise errors.AccessForbiddenError()
```

L'agent qui traite plus de 1000 demandes par semaine aura droit à une médaille.

#4 - 13 août 2019 15:43 - Frédéric Péters

> Est-ce qu'on ne devrait pas incrémenter uniquement sur un échec déjà ? Donc :

Je me suis posé la question et je me suis répondu non.

#5 - 13 août 2019 15:59 - Frédéric Péters

- Fichier 0001-misc-add-rate-limiting-to-tracking-code-URL-35386.patch ajouté

Voilà avec la valeur tirée du site-options.cfg et la possibilité de débrayer totalement.

#6 - 13 août 2019 16:29 - Benjamin Dauvergne

Avec 100 IPs trois essais par seconde, faut 25 jours pour trouver un code, avec 1 IP il faut presque 8 ans.

```
In [6]: 32**9/48633/(3*86400*7)/52/100.0*365
Out[6]: 25.55 # days

In [7]: 32**9/48633/(3*86400*7)/52.0
Out[7]: 7.653846153846154 # years
```

10000 par semaine (ou 1500 par jour) et par IP ça me vraiment parait plus sûr et suffisant :

```
In [8]: 32**9/48633/10000/52.0
Out[8]: 1391.2692307692307 # years

In [9]: 32**9/48633/10000/52.0/100
Out[9]: 13.912692307692307 # years
```

C'est en supposant en plus qu'on monte à 32 caractères et qu'on passe de 8 à 9 caractères, https://dev.entrouvert.org/issues/35118#note-19, avec les valeurs actuelles et 3req/s il ne faut toujours que 2j pour trouver un code :

```
In [9]: 20**8/48633/(3*86400.0)
Out[9]: 2.030829475308642
```

#7 - 13 août 2019 17:03 - Frédéric Péters

C'est fantastiquement une option de configuration.

#8 - 13 août 2019 17:06 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> 10000 par semaine (ou 1500 par jour) et par IP ça me vraiment parait plus sûr et suffisant

Yep. Sur un délai plus long de blocage il faudra disposer d'un outil de déblocage ou, au minimum, afficher clairement à l'usager qu'il y a une limitation atteinte (sinon ça sera bien délicat de faire du support).

Et je me dis qu'on devrait donc répondre explicitement, genre un raise errors.AccessForbiddenError('access is rate-limited') pour nous aider à pister l'affaire le jour où ça arrivera. Ack avec ça.

(Et oui, passer à 30**10 est de toute façon nécessaire, parce que rate-limit par IPv6, ha ha ha)

#9 - 13 août 2019 17:14 - Frédéric Péters

- Statut changé de Solution validée à Résolu (à déployer)

(message tapé sur le AccessForbidden)

```
commit 32f304fd511a7b80af6430d09ce3b006054e948c
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Tue Aug 13 13:58:55 2019 +0200

    misc: add rate limiting to tracking code URL (#35386)
```

#10 - 13 août 2019 17:24 - Benjamin Dauvergne

Frédéric Péters a écrit :

> C'est fantastiquement une option de configuration.

Je ne vois pas bien l'intérêt de mettre une valeur par défaut qu'on sait déjà ne pas être RGPD compliant, sachant qu'on doit l'être dans 100% des cas.

#11 - 13 août 2019 17:33 - Benjamin Dauvergne

Thomas Noël a écrit :

> Yep. Sur un délai plus long de blocage il faudra disposer d'un outil de déblocage ou, au minimum, afficher clairement à l'usager qu'il y a une limitation atteinte (sinon ça sera bien délicat de faire du support).

Si on ne comptait que les échecs c'est juste impossible que quelqu'un se retrouve bloqué par erreur, sauf DOS local sur une IP (genre utilisateur mobile qui décide de faire chier tous les gens sur sa passerelle NAT), ça limite grandement les cas.

(Et oui, passer à 30**10 est de toute façon nécessaire, parce que rate-limit par IPv6, ha ha ha)

django-ratelimit applique un netmask de 64 bits[1] pour IPv6 mais des idées que j'ai lu sur stackoverflow ce n'est pas suffisant certains ISP distribuant du /56 ou même du /48, idéalement il faudrait couvrir plusieurs netmask avec des compteurs de plus en plus large mais pas autant que l'augmentation du nombre d'IP.

[1] https://github.com/jsocol/django-ratelimit/blob/master/ratelimit/core.py#L32

#12 - 14 août 2019 11:15 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-misc-add-rate-limiting-to-tracking-code-URL-35386.patch	6,92 ko	13 août 2019	Frédéric Péters
0001-misc-add-rate-limiting-to-tracking-code-URL-35386.patch	7,7 ko	13 août 2019	Frédéric Péters